

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 9 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»

«Дослідження алгоритмів обходу масивів»

Варіант 3

Виконав студент ІП-13 Баран Софія Володимирівна
(шифр, прізвище, ім'я, по батькові)

Перевірів _____
(прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота 9
Дослідження алгоритмів обходу масивів
Варіант 3

*Мета-*дослідити алгоритми обходу масивів, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

Задано матрицю дійсних чисел $A[m, n]$. При обході матриці по рядках знайти в ній перший додатний елемент X і його місцезнаходження. Порівняти X із середньоарифметичним значенням елементів під побічною діагоналлю.

Розв'язання

Постановка задачі:

Результатом розв'язку задачі є змінна X (і її місцезнаходження) та її порівняння з Y , де X – перший додатний елемент, знайдений при обході заданої матриці $A[m, n]$ змійкою по рядках, а Y – середньоарифметичне значення елементів під побічною діагоналлю. Для заповнення матриці, її виведення, знаходження першого додатного елемента та обчислення середньоарифметичного використовуватимуться підпрограми.

Математична модель:

Змінна	Тип змінної	Ім'я	Призначення
Рядки	Цілий	m	Введене значення
Стовпці	Цілий	n	Введене значення
Матриця	Дійсний	matrix	Проміжне значення
#1 дод. елемент	Дійсний	X	Результат
Рядок ел. X	Цілий	$iMax$	Результат
Стовпець ел. X	Цілий	$jMax$	Результат
Сер. арифметичне	Дійсний	Y	Проміжне значення

Програмні специфікації запишемо у псевдокоді та у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо заповнення матриці, знаходження X та Y .

Псевдокод:

Крок 1

початок

int row = 6, col = 4;
заповнення matrix[m, n]
обчислення X та iMax, jMax
обчислення Y
порівняння X та Y

кінець

Крок 2

початок mtrx_crt(float[,] matrix)

Random rnd = new Random()

для (int i = 0; i < matrix.GetLength(0); i++)

для (int j = 0; j < matrix.GetLength(1); j++)

matrix[i, j] = (float) (rnd.Next(0, 100)/10.0)

все для

все для

повернення matrix;

кінець

початок mtrx_out(float[,] matrix)

для (int i = 0; i < matrix.GetLength(0); i++)

для (int j = 0; j < matrix.GetLength(1); j++)

виведення matrix[i,j]

все для

все для

кінець mtrx_out

початок posEl_fnd(float[,] matrix, out int iMax, out int jMax)

iMax = jMax = 0

для (int i = 0; i < matrix.GetLength(0); i++)

якщо (i % 2 == 0)

для (int j = 0; j < matrix.GetLength(1); j++)

якщо (matrix[i, j] > 0)

iMax = i

jMax = j

повернення matrix[i, j]

все якщо

все якщо

```

        інакше
            для (int j = matrix.GetLength(1) - 1; j > 0; j--)
                якщо (matrix[i, j] > 0)
                    iMax = i
                    jMax = j
                    повернення matrix[i, j]
            все якщо
        все інакше
    повернення 0
кінєць posEl_fnd

```

```

початок dgnl_avrg(float[,] matrix)
    float sum = 0
    int num = 0
    для (int i = 0; i < matrix.GetLength(0); i++)
        для (int j = 0; j < matrix.GetLength(1); j++)
            якщо ((i + j) >= matrix.GetLength(0))
                sum += matrix[i, j]
                num++
            все якщо
        все для
    все для
    повернення sum/num
кінєць dgnl_avrg

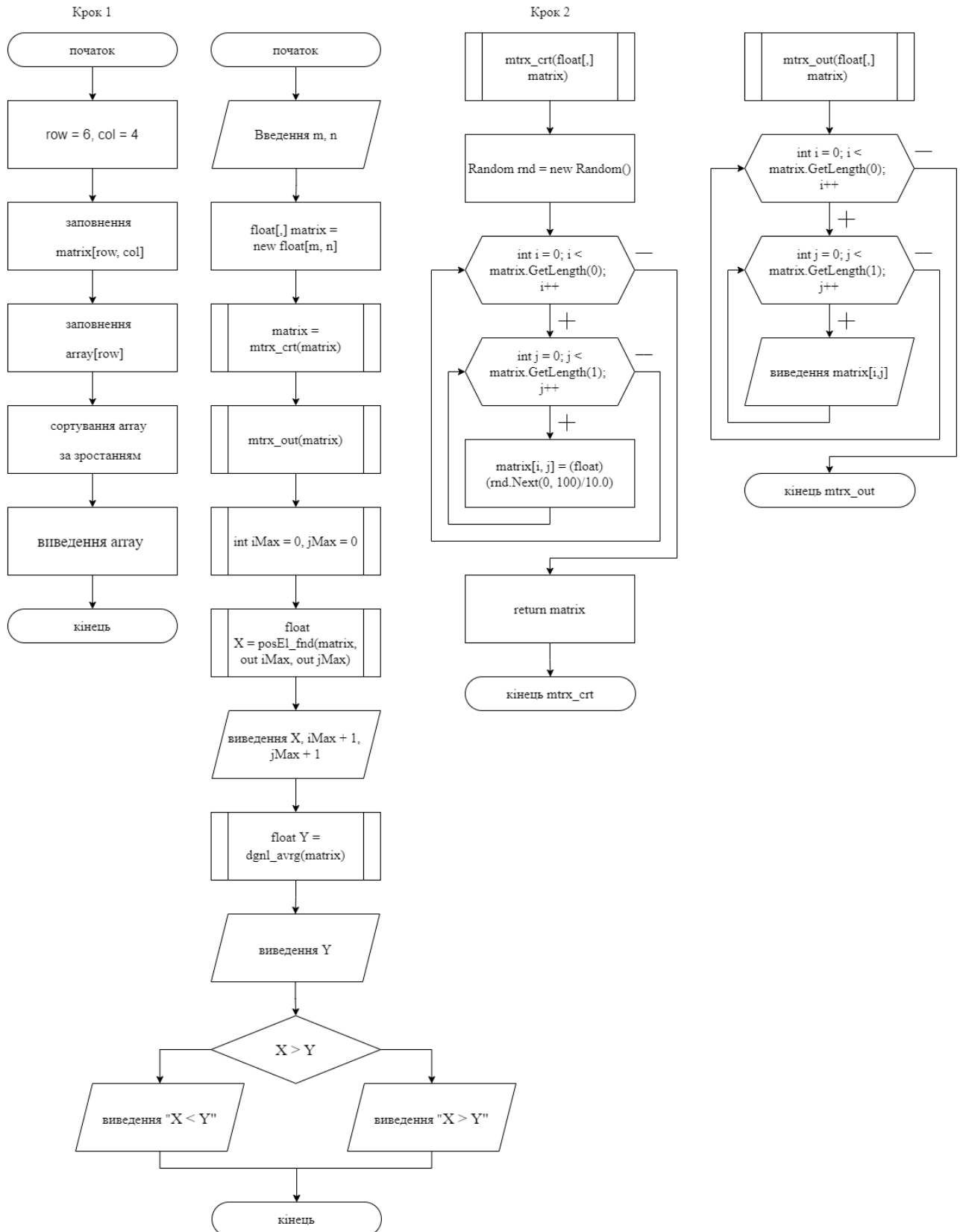
```

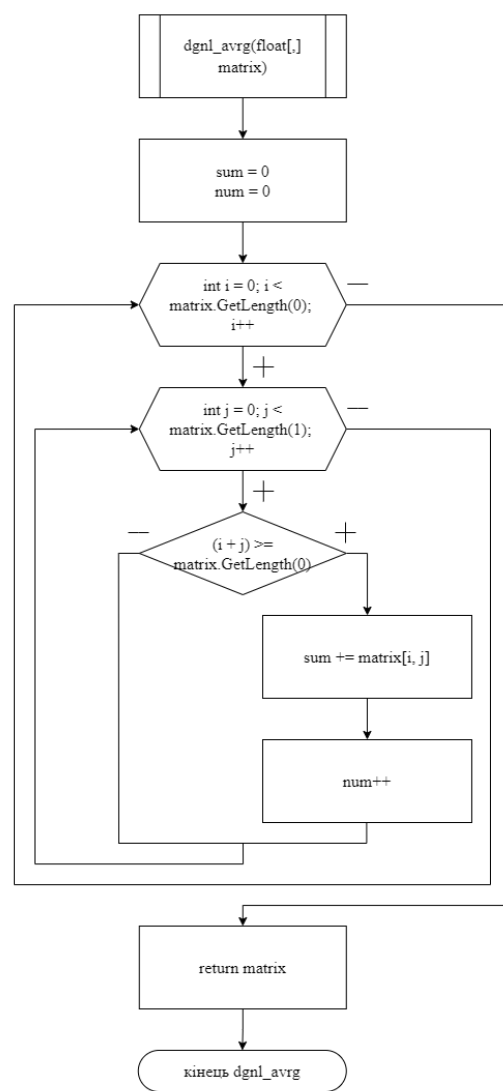
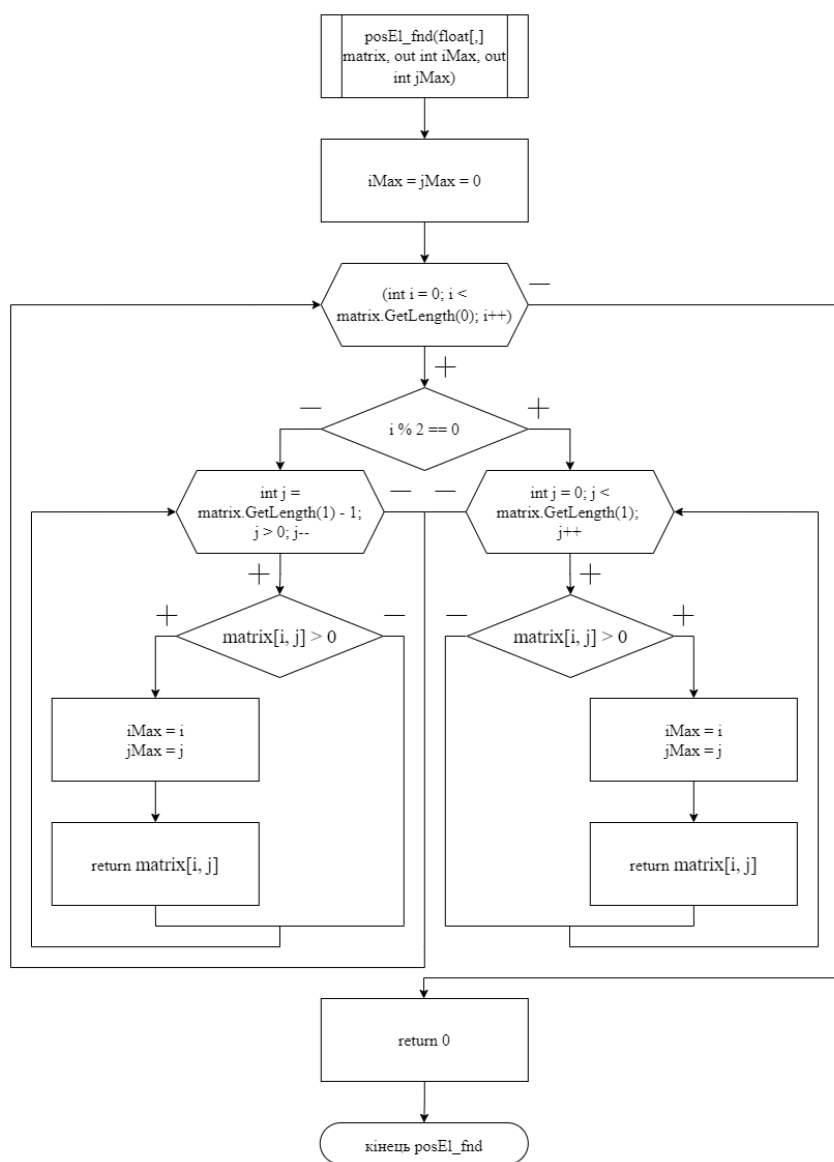
```

початок
    введення m, n
    float[,] matrix = new float[m, n]
    matrix = mtrx_crt(matrix)
    mtrx_out(matrix)
    int iMax = 0, jMax = 0
    float X = posEl_fnd(matrix, out iMax, out jMax)
    виведення X, iMax + 1, jMax + 1
    float Y = dgnl_avrg(matrix)
    виведення Y
    якщо (X > Y)
        виведення "X > Y"
    все якщо
        виведення "X < Y"
    інакше
кінєць

```

Блок-схема





Код програми (C#)

```
using System;
using System.Threading;

namespace Lab9ADS
{
    internal class Program
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("\nEnter size of a matrix.");
            Console.Write("Rows: ");
            int m = Convert.ToInt32(Console.ReadLine());
            Console.Write("Columns: ");
            int n = Convert.ToInt32(Console.ReadLine());
            float[,] matrix = new float[m, n];
            matrix = mtrx_crt(matrix);
            mtrx_out(matrix);
            int iMax = 0, jMax = 0;
            float X = posEl_fnd(matrix, out iMax, out jMax);
            Console.WriteLine($"The first positive element (X): {X} ({iMax + 1}, {jMax + 1})");
            float Y = dgnl_avrg(matrix);
            Console.WriteLine($"The average of side diagonal (Y): {Y}\n");
            Console.WriteLine(X > Y ? "X > Y" : "X < Y");
        }

        static float[,] mtrx_crt(float[,] matrix)
        {
            Random rnd = new Random();
            for (int i = 0; i < matrix.GetLength(0); i++)
                for (int j = 0; j < matrix.GetLength(1); j++)
                    matrix[i, j] = (float) (rnd.Next(-100, 100)/10.0);
            return matrix;
        }

        static void mtrx_out(float[,] matrix)
        {
            Console.WriteLine("\nGenerated matrix: ");
            for (int i = 0; i < matrix.GetLength(0); i++)
            {
                for (int j = 0; j < matrix.GetLength(1); j++)
                {
                    Console.Write($"{matrix[i, j]}\t");
                }
                Console.WriteLine();
            }
        }

        static float posEl_fnd(float[,] matrix, out int iMax, out int jMax)
        {
            iMax = jMax = 0;
            for (int i = 0; i < matrix.GetLength(0); i++)
            {
                if (i % 2 == 0)
                {
                    for (int j = 0; j < matrix.GetLength(1); j++)
                    {
                        if (matrix[i, j] > 0)
                        {
                            iMax = i;
                            jMax = j;
                            return matrix[i, j];
                        }
                    }
                }
                else
                {
                    for (int j = matrix.GetLength(1) - 1; j > 0; j--)
                    {
                        if (matrix[i, j] > 0)
                        {
                            iMax = i;
                            jMax = j;
                            return matrix[i, j];
                        }
                    }
                }
            }
        }
    }
}
```

```

        return 0;
    }

    static float dgn1 avrg(float[,] matrix)
    {
        float sum = 0;
        int num = 0;
        for (int i = 0; i < matrix.GetLength(0); i++)
        {
            for (int j = 0; j < matrix.GetLength(1); j++)
            {
                if ((i + j) >= matrix.GetLength(0))
                {
                    sum += matrix[i, j];
                    num++;
                }
            }
        }
        return sum/num;
    }
}

```

Перевірка коду

```

"C:\Program Files\JetBrains\JetBrains Rider
pa.detach.6792 C:/Users/MX1010A/RiderProject

Enter size of a matrix.
Rows: 3
Columns: 3

Generated matrix:
5.1      -8.9      6.6
4         -2.6      3.6
6.8       7.9       6.4

The first positive element (X): 5.1 (1, 1)

The average of side diagonal (Y): 5.966667

X < Y

Process finished with exit code 0.

```

```

"C:\Program Files\JetBrains\JetBrains Rider
pa.detach.6820 C:/Users/MX1010A/RiderProject

Enter size of a matrix.
Rows: 3
Columns: 3

Generated matrix:
-7.3      -5.4      4.3
-10        -6.4     -7.6
-0.8       3.8       8.8

The first positive element (X): 4.3 (1, 3)

The average of side diagonal (Y): 1.666667

X > Y

Process finished with exit code 0.

```

$$\frac{7,9+6,4+3,6}{3}$$

$$= \frac{179}{30}$$

Alternate form

$$5\frac{29}{30}, 5.9\dot{6}$$

$$\frac{3,8+8,8-7,6}{3}$$

$$= \frac{5}{3}$$

Alternate form

$$1\frac{2}{3}, 1.\dot{6}$$

Висновок: під час виконання завдання був досліджений алгоритм обходу масивів змійкою по рядках, крім того, були набуті практичні навички його використання під час складання програмних специфікацій, зокрема розв'язання задачі (що полягала у порівнянні змінної X та Y , де X – перший додатний елемент, знайдений при обході заданої матриці $A[m, n]$, а Y – середньоарифметичне значення елементів під побічною діагоналлю). Для складання алгоритму програми використовувалися псевдокод та блок-схеми, після чого його було реалізовано на компільованій мові програмування C# та перевірено його працездатність.